

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Absolvování individuální odborné praxe
Individual Professional Practice in the Company

Zadání bakalářské práce

Student:

Lukáš Satin

Studijní program:

B2647 Informační a komunikační technologie

Studijní obor:

2612R025 Informatika a výpočetní technika

Téma:

**Absolvování individuální odborné praxe
Individual Professional Practice in the Company**

Zásady pro vypracování:

1. Student vykoná individuální praxi ve firmě: Tieto Czech s.r.o.
2. Struktura závěrečné zprávy:
 - a) Popis odborného zaměření firmy, u které student vykonal odbornou praxi a popis pracovního zařazení studenta
 - b) Seznam úkolů zadaných studentovi v průběhu odborné praxe s vyjádřením jejich časové náročnosti
 - c) Zvolený postup řešení zadaných úkolů
 - d) Teoretické a praktické znalosti a dovednosti získané v průběhu studia uplatněné studentem v průběhu odborné praxe
 - e) Znalosti či dovednosti scházející studentovi v průběhu odborné praxe
 - f) Dosažené výsledky v průběhu odborné praxe a její celkové zhodnocení

Seznam doporučené odborné literatury:

Podle pokynů konzultanta, který vedl odbornou praxi studenta.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **RNDr. Eliška Ochodková, Ph.D.**

Konzultant bakalářské práce:

Lukáš Hejl

Datum zadání: 16.11.2012

Datum odevzdání: 07.05.2013



doc. Dr. Ing. Eduard Sojka
vedoucí katedry



prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlášení studenta

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně.
Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

Dne: 4.4.2013

Libuše Šedivá

Poděkování

Rád bych poděkoval týmu ComC za cenné rady v průběhu mé praxe a také RNDr. Elišce Ochodkové, Ph.D. za odbornou pomoc a konzultaci při vytváření této bakalářské práce.

Abstrakt

Cílem této bakalářské práce je popsat průběh mé praxe ve firmě Tieto Czech s.r.o. se zaměřením na popis řešení zadaných úkolů. Pracoval jsem na pozici Software Developer v oblasti Nordic Product Portfolio. Produkty v tomto portfoliu jsou založeny na technologiích Microsoftu. V první části je popsáno odborné zaměření firmy Tieto Czech s.r.o. a mé pracovní zařazení. Druhá část popisuje softwarovou architekturu systému. Část třetí pak úkoly, které jsem během praxe řešil. Poslední část se věnuje zhodnocení dosažených výsledků odborné praxe a zkušeností, které mi tato praxe přinesla.

Klíčová slova

Tieto, .NET Framework, Microsoft BizTalk, systémová integrace, integrace podnikových aplikací

Abstract

The aim of this bachelor thesis is to describe my experience during the individual professional practice in Tieto Czech s.r.o. Company. My position was a Software Developer in Nordic Product Portfolio. The actual products in this portfolio are based on Microsoft technologies. In the first chapter, Tieto company and my competence as a Software Developer is described. The second chapter provides some theoretical background required to perform my tasks. The third chapter describes actual tasks I was performing during the professional practice. Finally, the last chapter provides a conclusion and benefits of my work.

Key words

Tieto, .NET Framework, Microsoft BizTalk, system integrations, enterprise application integration

Seznam použitých symbolů, zkratek a termínů

ComC	Communication Center
EAI	Enterprise Application Integration
EDA	Event Driven Architecture
ESB	Enterprise Service Bus
ETL	Extract, Transform & Load
NPP	Nordic Product Portfolio
REST	Representational State Transfer
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
UBI	Utility Business Integrations
XML	Extensible Markup Language
XSD	XML Schema Definition

Obsah

1 Úvod.....	1
2 Odborné zaměření firmy.....	2
2.1 Tieto Czech s.r.o.....	2
2.2 Mé pracovní zařazení.....	2
3 Architektura projektu.....	3
3.1 Integrace systémů a podnikových aplikací.....	3
3.2 Propojení heterogenních systémů na transportní vrstvě.....	3
3.3 Druhy systémových integrací.....	6
4 Průběh praxe a řešené úkoly.....	7
4.1 ComC.....	7
4.2 UBI.....	8
5 Teoretické znalosti získané během studia uplatněné v průběhu odborné praxe.....	10
6 Znalosti a dovednosti scházející studentovi v průběhu odborné praxe.....	11
7 Dosažené výsledky v průběhu odborné praxe a její celkové zhodnocení.....	12
8 Použitá literatura.....	13

1 Úvod

Téma mé bakalářské práce jsem si zvolil na základě předchozí zkušenosti se společností Tieto Czech s.r.o., kdy jsem roku 2011 absolvoval kurz Tieto IT Academy – Java & Software development. Spolu s pracovním prostředím mě zaujaly především agilní metodiky vývoje. Po zaslání životopisu jsem byl kontaktován a pozván na přijímací pohovor k výběru na odbornou praxi. Na praxi jsem nastoupil v listopadu 2012. Byl jsem zařazen do týmu podílejícím se na vývoji v oblasti Nordic Product Portfolio. Produkty NPP jsou vyvíjeny na technologiích firmy Microsoft. Jedná se zejména o C# .NET Framework a Microsoft BizTalk.

V této práci, rozdělené do čtyř částí, budu popisovat průběh mé praxe v Tieto Czech s.r.o. V první části představím společnost a mé pracovní zařazení. Druhá část popisuje architekturu systému. Část třetí pak úkoly řešené v rámci praxe. Poslední část obsahuje zhodnocení dosažených výsledků a zkušeností získaných během praxe.

2 Odborné zaměření firmy

2.1 Tieto Czech s.r.o.

Tieto je největší severoevropský dodavatel IT služeb, který poskytuje komplexní služby pro soukromý i veřejný sektor. Společnost Tieto byla založena roku 1968 se sídlem v Helsinkách. Nyní zaměstnává 17 000 expertů a působí ve více než dvaceti zemích světa. Čisté tržby společnosti činí 1,8 miliard eur za rok. Akcie firmy jsou obchodovány na burze NASDAQ OMX v Helsinkách a Stockholmu. Roku 2001 společnost Tieto vstoupila také do České republiky a roku 2004 otevřela své softwarové centrum v Ostravě. V dnešní době, kdy se počet zaměstnanců společnosti pohybuje okolo 1900, je jedním z největších zaměstnavatelů v České republice a největším zaměstnavatelem v Moravskoslezském kraji. Z hlediska počtu kmenových zaměstnanců je česká pobočka třetí největší pobočkou Tieto korporace na světě. První dvě místa zauímají mateřské země Finsko a Švédsko [1].

2.2 Mé pracovní zařazení

Ve společnosti jsem vykonával praxi na pozici software developer v týmu ComC, později UBI. Oba týmy řeší integraci a komunikaci produktů v rámci NPP a také produktů třetích stran. Tým ComC vyvíjí produkt se stejným názvem, který slouží jako integrační platforma NPP a je vyvíjen v C# pod .NET Frameworkem. Tým UBI vyvíjí rovněž integrační platformu, ale založenou na Microsoft BizTalk Server 2010. ComC je interní produkt Tieto a historicky je zastoupen nejvíce ve Finsku. V ostatních zemích se nyní začíná více prosazovat platforma BizTalk, která slouží ke stejnému účelu. Zákazník má však v druhém případě svobodnější volbu ohledně subdodavatelů, protože BizTalk je produkt Microsoftu, který lze zakoupit i samostatně.

3 Architektura projektu

Projekt, jehož jsem byl součástí, se zabývá především systémovou integrací a integrací podnikových aplikací. S rostoucím počtem různých aplikací používaných v IT společnostech je kladen větší důraz na jejich vzájemnou integraci. Hlavní motivací je koordinovaná komunikace jednotlivých systémů nejen v rámci podniku, ale také mezi různými společnostmi. Produkty samotné nemusí splňovat veškeré požadavky zákazníka. Teprve vhodné propojení těchto produktů začne přinášet určitou hodnotu.

3.1 Integrace systémů a podnikových aplikací

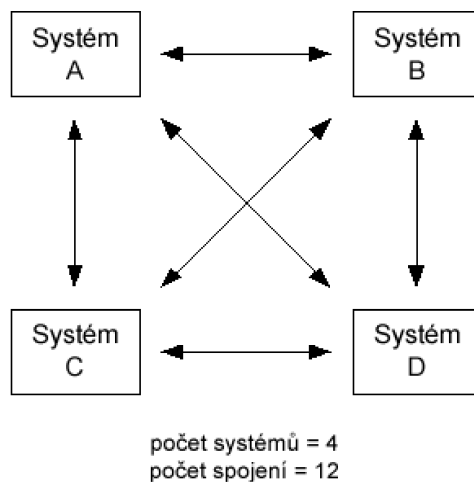
Systémovou integraci lze definovat jako [2]:

1. Technické propojování heterogenních systémů na nejrůznějších úrovních (hardware, protokoly, databázové systémy atd.);
2. Řešení jednoho konkrétního IS/IT projektu, přičemž systémovou integraci rozumíme optimální sladění projektového a právního přístupu tak, aby bylo dosaženo cílů projektu;
3. Dlouhodobý koordinovaný proces řešení IS/IT projektů vycházející ze strategického záměru zákazníka s cílem plnit jeho vyvíjející se požadavky na řízení procesů v organizaci;
4. Dlouhodobý koordinovaný proces řešení všech podnikových projektů (nejen IS/IT) s cílem naplňovat strategický záměr zákaznické organizace.

Integrace podnikových aplikací (EAI) je proces koordinované komunikace a propojení softwarových systémů v podniku. Většinou se jedná o snahu napojení zastaralých (legacy) aplikací a databází na nová rozhraní nebo kompletní modernizaci podnikové infrastruktury. Důraz je kladen na distribuovanou, multiplatformní komunikaci a spolupráci s webovými službami. EAI k naplnění cílů využívá prostředky jako objektově-orientované programování, komunikace pomocí standardizovaného XML formátu nebo distribuce zpráv pomocí agregace dat do fronty událostí [3].

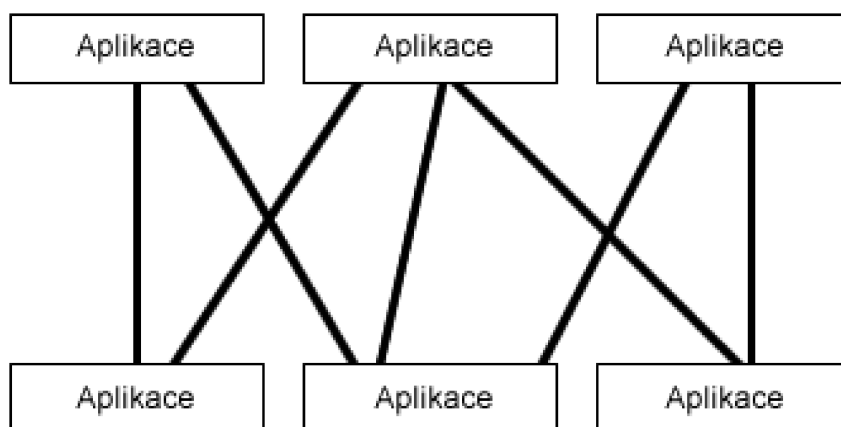
3.2 Propojení heterogenních systémů na transportní vrstvě

Složitost integrace závisí na počtu systémů. Nejjednodušším případem je propojení dvou heterogenních systémů pomocí transportní vrstvy, která bude zasílat zprávy pouze ze zdrojového do cílového umístění. Řada scénářů ale vyžaduje propojení více než dvou aplikací. U reálných projektů, kdy je integrováno například 10 aplikací, je zapotřebí vytvořit 45 až 90 propojení. Při malém počtu aplikací lze snadno přehlédnout, že se zde jedná o exponenciální nárůst počtu spojení (viz obrázek 3.1). Z hlediska závislosti aplikací jedné na druhé představuje toto těsnou vazbu [4].



Obrázek 3.1: Problém těsné vazby a počtu spojení.

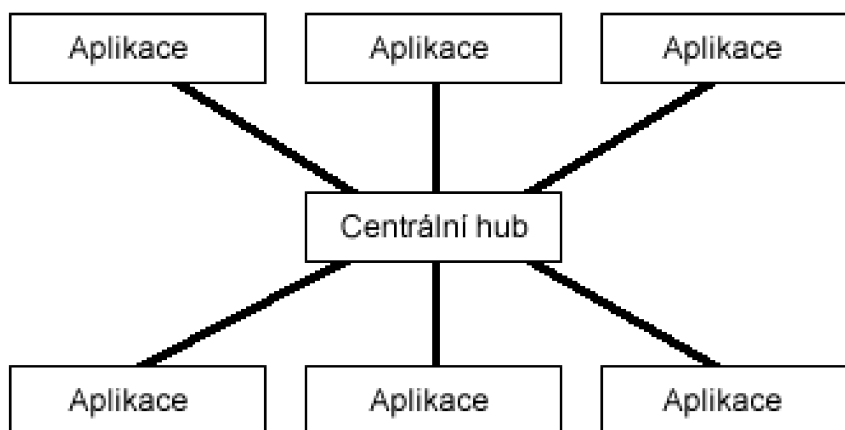
Propojení systémů v integrační vrstvě může mít vazbu těsnou nebo volnou. Těsná vazba se nazývá případ, kdy jsou jednotlivé systémy propojeny přímo a naprosto libovolně. Vzniká tak komplexní struktura, která je těžko spravovatelná. Naproti tomu vazba volná představuje ideální stav, kdy počet spojení nenarůstá exponenciálně a jednotlivé komponenty lze zaměnit. Jedním z nejdůležitějších pravidel správného objektově orientovaného programování je zásada programování proti rozhraní a ne proti implementaci. Nedodržení těchto zásad vede k návrhu s těsnou vazbou, což představuje ukázka Point-to-Point topologie na obrázku 3.2.



Obrázek 3.2: Point-to-Point topologie

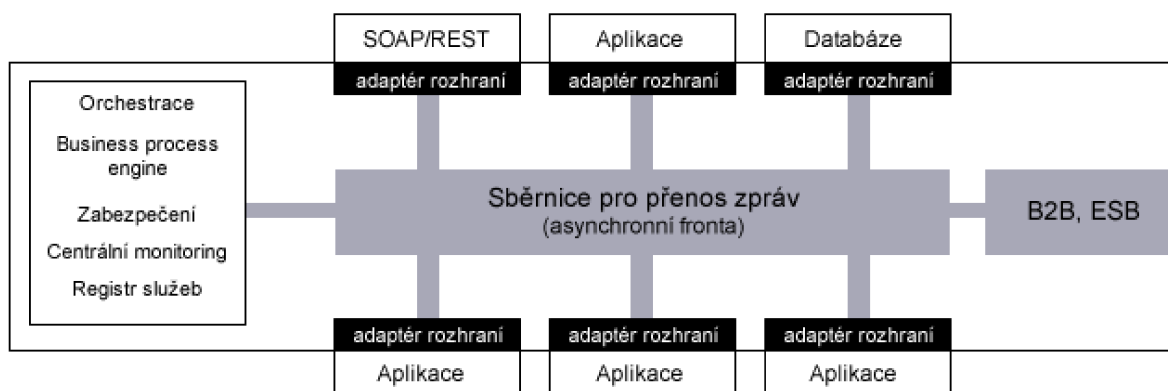
Mnohem lépe je na tom topologie Hub & Spoke (Obrázek 3.3), kdy uprostřed je umístěn centrální hub zajišťující veškerou komunikaci a obsahující v sobě veškerá rozhraní, takže zdrojový systém nemusí znát rozhraní cílového systému a naopak. Tato topologie představuje již vazbu volnou a

výrazně snižuje počet propojení mezi komponentami. Umožňuje též snadné monitorování komunikace. Nevýhodou je, že tento centrální hub je nejkritičtější místem celé infrastruktury jak z pohledu výkonu, tak z pohledu spolehlivosti.



Obrázek 3.3: Hub & Spoke topologie

Nejmodernější poznatky z oboru představují sběrníkovou topologii Enterprise Service Bus (Obrázek 3.4). Jedná se o distribuované řešení, které má více přípojných bodů. Komunikace mezi systémy je standardizovaná pomocí kanonického XML formátu. Veškerý provoz je třeba tedy převést do tohoto specifického XML formátu obsahujícího rovněž XSD schéma pro validaci datových typů. Následně lze již použít řadu generických komponent pro transformaci dat a ušetřit tak čas, který by byl potřeba k ruční implementaci těchto komponent. Sběrnice jsou založeny na systému předávání zpráv (message-oriented middleware), což poskytuje volnou vazbu mezi komponentami. Komunikující systémy nemusí být funkční ve stejnou dobu, protože tyto zprávy se ukládají do samostatné databáze.



Obrázek 3.4: Enterprise Service Bus

3.3 Druhy systémových integrací

Jednotlivé druhy se liší použitými návrhovými vzory a technologiemi. Pro různé účely jsou vhodné různé typy integrací.

- **Servisně orientovaná architektura (SOA)**

Integrace pomocí služeb slouží nejčastěji k integraci systémů s uživatelským rozhraním nebo pro integraci procesů běžících v reálném čase. Typický případ použití je synchronní volání s odpovědí. Komunikace systémů probíhá v reálném čase s důrazem na rychlost odezvy [5].

- **Událostmi řízená architektura (Event Driver Architecture, EDA)**

Tento typ integrace je vhodný pro jednosměrný přenos zpráv mezi systémy a synchronizace dat v reálném čase. Klíčové je doručení událostí i systémům, které nejsou v danou chvíli online [6].

- **Datová integrace (Extract-Transform-Load, ETL)**

Datová integrace řeší zejména pravidelný přenos dat, jejich transformaci a následné uložení [7]. Mezi základní funkcionalitu patří nástroje k monitorování průběhu a výsledku zpracování velkého objemu dat. Tyto úlohy probíhají často v pravidelných dávkách (např. 1 denně).

Komplexní firemní infrastruktura v praxi vyžaduje správnou implementaci všech tří, výše uvedených, druhů. Modernější řešení tak funguje například na principu SOA-EDA, kdy EDA rozšiřuje původní SOA architekturu právě o používání událostí. Bez událostí by bylo potřeba neustále v cyklu kontrolovat změny jednotlivých stavů, což zbytečně zatěžuje procesor a snižuje výkon.

4 Průběh praxe a řešené úkoly

Před nástupem na samotnou praxi jsem absolvoval dvoudenní školení spolu s nově přijatými zaměstnanci firmy. Hlavní náplní byly základní informace o společnosti a vnitřních procesech. Součástí byl též písemný test z anglického jazyka sloužící k rozdělení do skupin pro další rozvoj jazykových prostředků. Vzhledem k dosažené úrovni C1 [8] by pro mě největším přínosem byly hodiny s rodilým mluvčím se zaměřením na správnou výslovnost a slovní zásobu.

První den na odborné praxi jsem dostal počítač a veškeré potřebné zázemí k výkonu praxe. Line manager mě provedl po celém patře a alespoň zběžně jsem poznal členy ostatních týmu. Má cesta skončila u týmu ComC, který zajišťuje integraci a optimální komunikaci mezi jednotlivými produkty NPP. Zde jsem dostal úvodní školení týkající se mého zařazení v týmu a konkrétní podoby softwarové architektury ComC. S teoretickými znalostmi z předchozí kapitoly a konkrétní implementací ComC jsem již mohl začít vykonávat první úkoly.

4.1 ComC

Než jsem mohl začít řešit každodenní úkoly z plánovacího software, musel jsem vykonat praktické cvičení v podobě doprogramování C# kódu pro definici komponenty v konkrétní integraci. Tato komponenta měla načíst data ze SOAP webové služby a výsledek uložit do XML souboru na disku. Po implementaci a otestování komponenty ve vývojovém prostředí Visual Studio 2010 bylo dalším úkolem nasazení této integrace na testovací klon reálného zákaznickova prostředí. Konkrétní prostředí může obsahovat např. čtyři virtuální stroje Windows Server 2008 R2, přičemž na každém běží jiná aplikace. Dohromady toto modulární řešení tvoří jeden celek přinášející zákazníkovi určitou hodnotu. Důležitou vlastností je náhrada jakékoliv komponenty v systémové integraci. Jednotlivé firmy z různých zemí jsou totiž napojeny na různá rozhraní svých subdodavatelů. Splněním tohoto úkolu jsem si vyzkoušel integraci Hub & Spoke architektury v praxi.

Po vykonání praktického cvičení v podobě první implementace jsem se začal účastnit denních Scrum meetingů, kde mi byly přidělovány úkoly dle aktuálních priorit. Ať už se jednalo o zjištění příčiny a následné odstranění závady v konkrétní integraci pro určitého zákazníka nebo jen úprava interního dokumentu, postup byl vždy stejný. Práce probíhala dle metodiky Scrum, což znamená, že zadání úkolu obsahovalo:

- **User Story (US)**

Obsahuje popis problému v podobě jedné či více vět běžnou řečí. Odpovídá na otázky: Kdo? Co? Proč?

- **Definition of Done (DoD)**

Seznam dílčích úkolů včetně časové náročnosti, které vedou k úspěšnému řešení. Často zahrnuje i revizi řešení s jiným členem týmu či týmové mini školení.

- **Acceptance Criteria (AC)**

Popis, co musí být splněno pro uzavření úkolu. Oproti User Story je zde již odborněji popsán stav popisující fungující situaci.

Pokud byl v software nějaký defekt, jednalo se o úkol typu Problem report. Po analýze zadání jsem v plánovacím software vytvořil dílčí úkoly, kdy se vždy jednalo o trojici Investigation, Fix a Test.

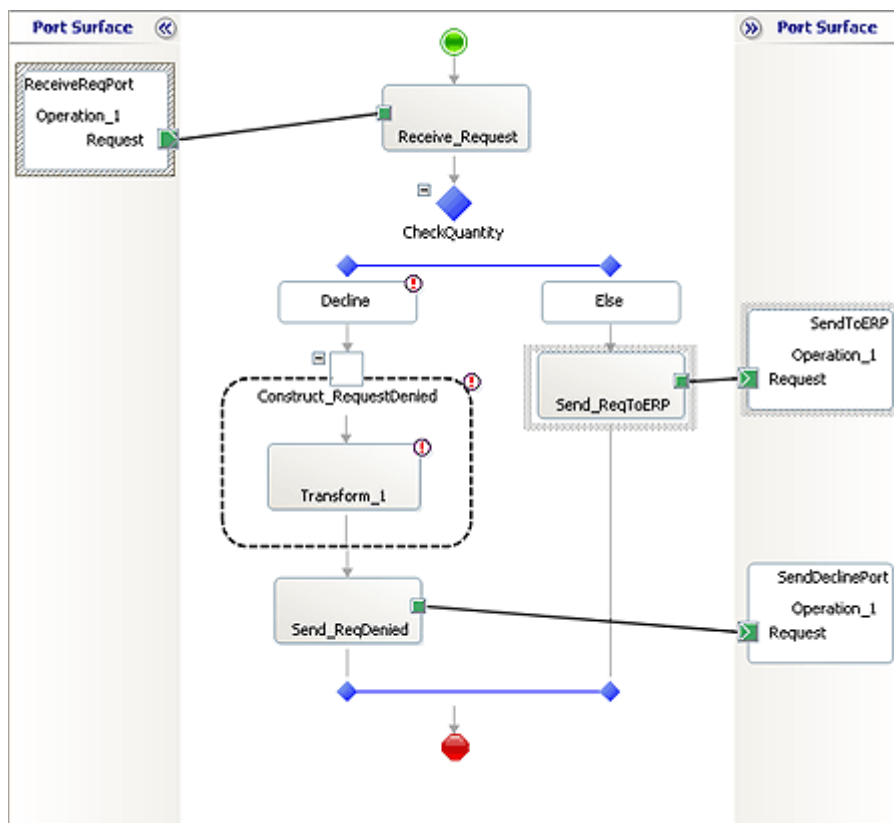
Nejprve jsem analyzoval problém a zjistil za jakých podmínek nastane nefunkční stav či výjimka. Souběžně s tímto jsem musel kontaktovat část týmu sídlící ve Finsku a zjistit do které verze má jít případná oprava. Podrobnosti jsem zaznamenal do dílčího úkolu Investigation a ten pak v plánovacím software uzavřel. Poté jsem mohl přejít k samotné opravě problému, což v sobě zpravidla zahrnovalo C# a XML kód. Poslední dílčí úkol s názvem Test měl již na starost kolega tester. V této době byl kód otestován ve vývojovém prostředí a většinou také na testovacím klonu v případě virtuálního stroje. Tester toto otestoval ještě důkladněji a na jiném testovacím klonu s databází jiného zákazníka.

Členy týmu jsem se snažil obtěžovat co nejméně, protože měli kromě podobné práce ještě řadu administrativních procesů. Během mé praxe přecházeli někteří členové týmu do týmu UBI, takže jsem tušil, že mě tento přechod také bude čekat.

4.2 UBI

Koncem mé praxe jsem měl možnost zaznamenat přechod z týmu ComC do týmu UBI. Zde se mi hodila již praktická znalost jednotlivých rozhraní a integrací z předchozích úkolů. Tým UBI implementuje totožnou funkcionalitu, jen místo interního Tieto produktu je použit produkt Microsoft BizTalk. Hlavní výhoda tohoto řešení představuje možnost zákazníka objednat si BizTalk konzultaci či implementaci u různých subdodavatelů. Architektura tohoto projektu není již Hub & Spoke, ale jedná se o plnohodnotný Enterprise Service Bus se širokými možnostmi pro audit a business reporting. Téměř nic se zde již neimplementuje v C#. K vývoji slouží rovněž Visual Studio. Jednotlivé procesy v integraci se zde ale definují graficky, což vede k lepší abstrakci zdrojového kódu na business procesy. Tato skutečnost by měla vést ke zlepšení komunikace mezi IT odborníky a vyšším managementem, který definuje své potřeby a business procesy.

Aktuální práce se zaměřuje především na převod stávajících integrací do Microsoft Biztalk. Nejvýhodnější je použít již předpřipravené generické komponenty pro transformaci nebo posloupnost dílčích operací definovaných v grafickém editoru ve Visual Studiu. Toto ovšem nelze vždy, někdy je potřeba doprogramovat část C# kódu či skriptu a zapouzdřit danou část jako komponentu. Ideální je přidat popisný název, aby hned ze schématu orchestrace šlo poznat, co daný kód provádí. Ukázka takového příkladu je k vidění na obrázku 4.1.



Obrázek 4.1: Grafický editor BizTalk orchestrací ve Visual Studio 2010 [9]

5 Teoretické znalosti získané během studia uplatněné v průběhu odborné praxe

Během studia jsem již implementoval ESB architekturu v Javě, což se mi u této odborné praxe velice hodilo. Z vyučovaných předmětů považuji v tomto případě za nejvíce přínosné předměty Architektura technologie .NET a Databázové a informační systémy. Získal jsem především praktické zkušenosti s prací ve vícečlenném týmu. Někdy bylo potřeba analyzovat problém s členy ostatních týmů, zejména pokud se jednalo o integraci s ostatními produkty Tieto a změnu v aplikačním rozhraní nějaké služby. Za nejprínosnější vlastnost považuji systematický a logicky správný postup při řešení dílčích úloh, což se neobejde bez teoretických základů objektově orientovaného programování. Samotná znalost teorie, ale nestačí. Je potřeba vše prakticky vyzkoušet ještě před ostrou implementací pro zákazníka. Čas vždy dané řešení prověří a odkryje případné nedostatky plynoucí z neustále se měnících požadavků.

6 Znalosti a dovednosti scházející studentovi v průběhu odborné praxe

Vzhledem ke skutečnosti, že programuji od dětství a od svých 15-ti let realizuji praxi v různých firmách ve formě dohody o provedení práce, jsem neměl při vykonávání této odborné praxe žádné problémy. Musel jsem se seznámit hlavně s konkrétním schématem databáze a konkrétní implementací dané architektury.

7 Dosažené výsledky v průběhu odborné praxe a její celkové zhodnocení

Odbornou praxi hodnotím jako velice přínosnou. Získal jsem praktické zkušenosti s prací v mezinárodním týmu, kde komunikace v angličtině probíhá denně. Jsem rád, že jsem zde byl také v době, kdy se přecházelo na novou integrační platformu Microsoft Biztalk a měl jsem možnost získat zkušenosti v obou týmech ComC/UBI. Snažil jsem se vždy být součástí týmu a pomáhat v podobě volné kapacity pro plnění úloh.

8 Použitá literatura

- [1] TIETO. *Informace o Tieto* [online]. 2013 [cit. 2013-03-27]. Dostupné z: <http://www.tieto.cz/o-nas>
- [2] HABÁŇ, Jaromír a Petr SODOMKA. *Systems integration 2004: 12th international conference, Prague, Czech Republic, June 14-15, 2004 : proceedings* [online]. Ed. 1. Praha: Vysoká škola ekonomická, 2004 [cit. 2013-03-27]. 584 s. ISBN 80-245-0701-3. Dostupné z: <http://si.vse.cz/archive/proceedings/2004/jak-rozumi-systemove-integraci-dodavatele-informacnich-systemu.pdf>. D - Článek ve sborníku. Vysoká škola ekonomická v Praze.
- [3] What is EAI (enterprise application integration)?. In: *EAI (enterprise application integration)* [online]. 2007 [cit. 2013-03-28]. Dostupné z: <http://searchsoa.techtarget.com/definition/EAI>
- [4] KAYE, Doug. *Loosely Coupled: The Missing Pieces of Web Services* [online]. 2003 [cit. 2013-03-28]. Dostupné z: <http://searchsoa.techtarget.com/news/887229/Loosely-Coupled-The-Missing-Pieces-of-Web-Services>
- [5] ZIMMERLI, Brayan. *Business Benefits of SOA* [online]. University of Applied Science of Northwestern Switzerland, School of Business, 2009 [cit. 2013-03-27]. Dostupné z: <http://www.brayan.com/projects/BenefitsOfSOA/default.htm>
- [6] MICHELSON, Brenda. Event-Driven Architecture Overview. In: *Event-Driven Architecture Overview* [online]. Patricia Seybold Group, 2006 [cit. 2013-03-27]. DOI: 10.1571/bda2-2-06cc. Dostupné z: <http://www.customers.com/articles/event-driven-architecture-overview/>
- [7] Data Movement Patterns. *Data Movement Patterns* [online]. 2003 [cit. 2013-03-28]. Dostupné z: <http://msdn.microsoft.com/en-us/library/ff650072.aspx>
- [8] Common European Framework of Reference. *Common European Framework of Reference* [online]. 2008 [cit. 2013-03-31]. Dostupné z: <https://www.teachers.cambridgeesol.org/ts/exams/CEFR/C1>
- [9] Tutorial 1: Enterprise Application Integration. *Tutorial 1: Enterprise Application Integration* [online]. 2009 [cit. 2013-04-03]. Dostupné z: <http://msdn.microsoft.com/en-us/library/aa547974%28v=bts.10%29.aspx>